Classification of comments on social media based on long short-term memory

Chenyue Qiu

Multimedia, Mobile and Web Development, Fuzhou University, Fuzhou, 350000, China

832204112@fzu.edu.cn

Abstract. Social media has assumed a pivotal role in contemporary society, significantly enhancing the convenience of daily lives. Nonetheless, the prevalence of toxic comments on social media platforms has led to varying degrees of harm for individuals. The conventional practice of manually categorizing and blocking such toxic comments has proven to be highly inefficient. To address this issue, this study employs artificial intelligence natural language processing technology to classify social media comments, offering a more effective solution. In the past few years, many algorithms for handling text classification tasks have been introduced and applied in various scenarios. In this work, the author used an LSTM model that can effectively handle long sequence dependency problems to implement text classification. This study achieved an accuracy of 99.4% after training on the Kaggle toxic comments datasets. During the training process, the training accuracy is greater than the validation accuracy while the validation loss is lower than the training loss. After training, the trained model can accurately predict an input sentence and the results are within the expected range.

Keywords: Natural Language Processing, Text classification, Long Short-Term Memory.

1. Introduction

As the internet continues to evolve and grow, social media has emerged as a primary means for individuals to gain insights into their surroundings and express themselves. Users are able to share their views and opinions on social media. Although this facilitates people's lives and improves people's quality of life, unfortunately, social media also harbors a significant presence of negative and harmful comments, leading to cyberbullying and harassment that can inflict psychological harm on individuals [1]. A study demonstrated that 41% of US internet users have experienced cyberbullying and 61% have witnessed such behavior [2]. Many social media platforms have developed new technologies to classify comments in order to limit toxic comments.

An effective way to solve this problem is to employ Artificial Intelligence (AI) and Machine Learning (ML) to achieve text classification. In the past several years, artificial intelligence has found applications across a multitude of domains, encompassing Deep Learning, Computer Vision, and Natural Language Processing and intelligent robots, among others. Text classification has become a hot topic in NLP. The use of text categorization approaches in applications is becoming more and more popular among scholars, especially in light of recent developments in NLP and text mining [3]. It is developing rapidly and become more and more mature [4]. Many industries, including target marketing, medical diagnosis, news group filtering, document organization, datasets analyzing, machine learning, databases, and retrieving information groups have employed text categorization [5]. Most text classification and document categorization schemes can be classified as the following four phases: Evaluations, selecting the most effective classifiers, and feature extraction [6].

Over the last few decades, a variety of models for text classification have been suggested [6]. For traditional models, the first model applied to the text categorization job is Naïve Bayes Classifier (NB) Algorithm. Then generic classifiers—also referred to as classification models—that are often employed for text categorization are suggested, including the K-Nearest Neighbors (KNN) algorithm, Support Vector Machine (SVM), and Random Forest (RF)[3]. Recently, in the field of natural language processing, many new deep learning models and algorithms have appeared. The Transformer model proposed by Google completely avoids recurrence in favor of drawing global dependencies between input and output [7]. Bidirectional Encoder Representations from Transformers (BERT) mitigates the need for unidirectionality by employing a pre-training objective known as the "Masked Language Model" (MLM) [8]. Long-term dependencies in the data are easier to spot and use using the Long Short-Term Memory (LSTM) paradigm [9].

The harmful impact of toxic comments on social media has garnered significant national and societal concern. Effectively addressing the classification and containment of these comments is now an urgent priority. Hence, the principal objective of this study is to employ the Long Short-Term Memory (LSTM) model in the field of natural language processing for the purpose of categorizing comments shared on social media. The datasets was sourced from the Kaggle website, a popular platform for data science and machine learning enthusiasts to share and access datasets for various research and analytical purposes. A large portion of the Wikipedia comments in the dataset have been flagged as dangerous by human raters. Toxic, severely toxic, obscene, threatening, insulting, and identity-based hate are the several sorts of toxicity. The goal of this effort is to use the model developed to forecast the likelihood of each comment's toxicity.

2. Methods

2.1. Data preparation

This study used Wikipedia comments that were flagged by human raters as having harmful conduct on the Kaggle website. The original file contains four csv files which are train.csv, test.csv, sample_submission.csv and test_labels.csv. The author exclusively utilizes the "train.csv" file and employs a convenient approach by partitioning 20% of the "train.csv" dataset for validation purposes. Within the original "train.csv" dataset, there exist a total of 159,571 comments, each of which is associated with six distinct categories: toxic, severe toxic, obscene, threat, insult, and identity_hate, with binary labels assigned to each category.

The data preparation includes two parts. Firstly, clean and normalize text data. Secondly, tokenize the text by turning it into sequences of numbers, and ensure that all sequences possess identical lengths by padding or truncating the ends of any longer sequences. The first part has 5 steps. 1. Making the text case-insensitive by changing all of the characters in the text to lowercase. 2. Using regular expressions to eliminate any non-word and non-whitespace characters (punctuation symbols and special characters), URLs (strings that begin with "http" followed by non-whitespace characters), and all numerals (0-9) from the text. 3. Using a translation table produced by string.punctuation to remove all punctuation characters using the "translate" technique. 4. Removing any leading or following whitespace before replacing the newline (n) and tab (t) characters with spaces. 5. Whitespace should be used to separate the words in the text, and then each word should be joined back together with a single space. The second part involves 3 steps. 1. Building a vocabulary based on the frequency of terms in the training data while keeping the top 100,000 most common words in the data. 2.Converting the training sentences and validation sentence into sequences of integers. 3. Padding or truncating the training sequences to a fixed length of 300 tokens. All these steps are necessary for feeding them into a neural network.

2.2. LSTM model

This project predominantly employs an LSTM model to construct the neural network. LSTMs are chosen for their ability to effectively integrate previous information, making them especially suitable for handling time series data. Additionally, the utilization of neural network ensembles is incorporated to reduce result variability and improve generalization in the model's performance [10]. It has been applied in Handwriting recognition, machine translation and image processing etc. [11]. Before the LSTM model was proposed, the traditional RNN model was prone to “vanishing gradient” problems, limiting their ability to capture long-range dependencies in data. An advanced variety of RNN called LSTM was created to deal with this problem. They include a more intricate internal structure, such as gates that regulate the information flow inside the network. These gates enable LSTM to selectively recall or forget information from earlier time steps, which makes LSTM suited for modeling long-range dependencies in sequential data [12].

The author introduces a straightforward sequential neural network designed for text processing, comprising three distinct layers. The network commences with an embedding layer tasked with transforming integer-encoded words, generated during the tokenization step, into compact vectors of consistent dimensions. Subsequently, an LSTM layer is employed, featuring 16 neurons and employing the hyperbolic tangent (tanh) activation function to capture sequential information effectively. Finally, the network culminates in a dense layer equipped with six output units, serving the ultimate purpose of classification or prediction. This layer employs the "sigmoid" activation function, which is frequently applied to issues involving binary classification or multi-label classification, where each output unit relates to a binary classification task.

2.3. Implementation details

In this project, the author specifies three important aspects of training: loss function, optimizer and evaluation metrics. The loss function was set to binary cross-entropy. This loss function is commonly used for binary classification problems. The Adaptive Moment Estimation (Adam) optimizer widely used in many studies [13, 14], a well-liked optimization technique renowned for its effectiveness in training deep neural networks, was selected as the optimizer. The evaluation metrics were set to accuracy to track the accuracy metric during training. As the Adam optimizer dynamically adapts the learning rate throughout the training process by considering the gradients of the loss function concerning the model's parameters, this project didn't set the learning rate. The learning rate for each parameter is adjusted using a variety of methods, including momentum and adaptive scaling. Throughout the training process, the model utilizes the Adam optimizer to minimize the binary cross-entropy loss and optimize the model's weights and biases. Additionally, at the conclusion of each epoch, it computes the accuracy metrics for both the training and validation datasets.

3. Results and discussion

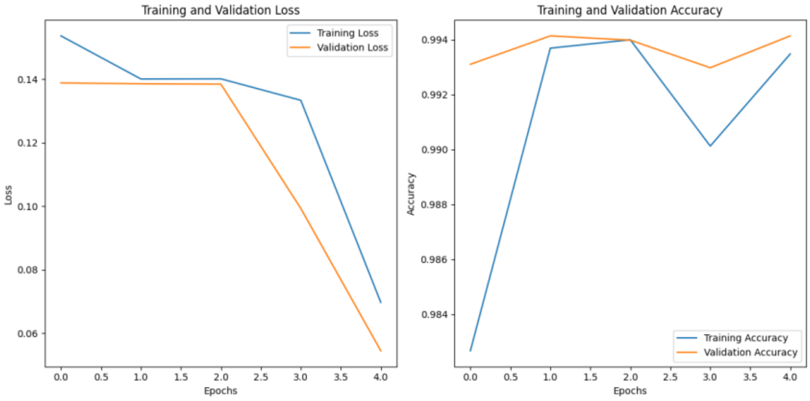

Figure 1. The loss and accuracy during the training process (Photo/Picture credit: Original)

After adjusting the loss function, optimizer and evaluation metrics, this study trained the model for 5 epochs and used the history function to record the training loss, validation loss, training accuracy and validation accuracy of each epoch. This study used the plot function to plot the loss and accuracy of each epoch. It can be seen in Figure 1. As depicted in Figure 1, it is evident that as the training epoch approaches 5, both the training loss and the validation loss show a constant decrease trend as the training epoch approaches 5. Notably, the validation loss continuously stays lower than the training loss over this time. When the training epoch reaches 5, the validation accuracy is almost equal to 99.4% and the training accuracy is almost always lower than the validation accuracy throughout the process. All the above show good signs.

To display the model's performance outcomes following training, this study defined an input sentence for the model to predict it. The input sentence was “You are a freak, I am going to hit you.” The final prediction result is six probability values corresponding to six category labels. The output of the prediction is shown in Table1.

Table 1. The prediction output

categories	Toxic	Severely_toxic	Obscene	Threatening	Insulting	Identity_hate
probability	0.9327658	0.18326572	0.8282436	0.04570375	0.7650934	0.16787836

According to Table 1, it can be observed that the three categories with the largest probabilities in the results predicted by the model are toxic, observation and insult.In the results, the probability of obscene is somewhat high. The possible reason is that the various categories of data are not balanced during the data processing process. However, the predicted results are generally impressive. This finding somewhat supports the LSTM model's strong performance on text categorization tasks.

Although this study achieved the expected results, there are still some limitations. The data set collected in this study was social media comments from five years ago, and with the rapid development of social media, toxic comments are constantly changing. Therefore, the model trained using these datasets may not be able to accurately predict current social media comments. In the future, datasets will need to be continuously updated to adapt to the evolving society.

4. Conclusion

The study used natural language processing in artificial intelligence to classify toxic comments in social media. In this research, the author preprocessed the datasets and then built an LSTM model to train the datasets. In the end, the trained model is used to predict an input sentence and the prediction results are as expected. By analyzing the prediction results and loss and accuracy curves after multiple experiments, it can be proved that the LSTM model performs well in text classification tasks. The possible limitation in this study is that the data set in this study comes from social media comments from five years ago and may not accurately predict new toxic comments emerging today. In the future, the author plans to update the datasets to adapt to the current social media environment, and balance the datasets during data processing to make its results more accurate.

References

[1] Spiros V et al 2018 Convolutional Neural Networks for Toxic Comment Classification. In Proceedings of the 10th Hellenic Conference on Artificial Intelligence (SETN '18) Association for Computing Machinery New York NY USA Article 35 1–6

[2] Wikipedia: No personal attacks", Wikipedia, [online] Available: https://en.wikipedia.org/wiki/Wikipedia:No_personal_attacks.

[3] Li Q Peng H Li J et al. 2020 A survey on text classification: From shallow to deep learning arXiv preprint arXiv:2008.00364

[4] Myagmarsuren O et al 2023 Performance analysis of a novel hybrid deep learning approach in classification of quality-related English text

[5] Aggarwal C C Zhai C X 2012 A survey of text classification algorithms Mining text data 163-222

[6] Kowsari K Jafari Meimandi K Heidarysafa M et al. 2019 Text classification algorithms: A survey Information 10(4): 150

[7] Vaswani A Shazeer N Parmar N et al. 2017 Attention is all you need Advances in neural information processing systems 30

[8] Devlin J Chang M W Lee K et al. 2018 Bert: Pre-training of deep bidirectional transformers for language understanding arXiv preprint arXiv:1810.04805

[9] Huang Z Xu W Yu K 2015 Bidirectional LSTM-CRF models for sequence tagging arXiv preprint arXiv:1508.01991

[10] Fjellström C 2022 Long short-term memory neural network for financial time series 2022 IEEE International Conference on Big Data (Big Data) IEEE 3496-3504.

[11] Staudemeyer R C Morris E R 2019 Understanding LSTM--a tutorial into long short-term memory recurrent neural networks arXiv preprint arXiv:1909.09586

[12] Sherstinsky A 2020 Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network Physica D: Nonlinear Phenomena 404: 132306.

[13] Qiu Y et al 2020 Improved denoising autoencoder for maritime image denoising and semantic segmentation of USV China Communications 17(3) 46-57.

[14] Mehta S 2019 CNN based traffic sign classification using Adam optimizer. In 2019 international conference on intelligent computing and control systems (ICCS) pp. 1293-1298 IEEE.